UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/030,258	02/25/1998	RICHARD KENNETH SCHULTZ	12217-100	7125

27267 7590 02/12/2004

WIGGIN & DANA LLP
ATTENTION: PATENT DOCKETING
ONE CENTURY TOWER, P.O. BOX 1832
NEW HAVEN, CT 06508-1832

EXAMINER	
LAO, SUE X	
ART UNIT	**PAPER NUMBER**
2126	31

DATE MAILED: 02/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/030,258	SCHULTZ ET AL.
	Examiner	Art Unit
	S. Lao	2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 November 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-46 and 60-63* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-46 and 60-63* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-46 and 60-63 are pending. This action is in response to the amendment filed 11/26/2003 accompanying an RCE. Applicant has amended claims 1, 33, 34, 60, 62 and 63.

2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

3. Claim 34 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim34 recites "one or more instructions" on line 4 and "a next instruction within said selected script" on lines 17-18, which are conflicting in that the next instruction would not exist if there is only one instruction in the selected script. For the purpose of art rejection, the "one or more instructions" on line 4 is interpreted as "more than one instructions", as best understood and as it appears to be.

4. Claims 1-46 and 60, 61, 63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Skillman et al (US Pat. 5,506,999) in view of Fischer (US Pat. 5,337,360).

As to claim 1, Skillman teaches a plurality of processing units each executing a portion of an overall application and a centralized event-driven control to integrate the distributed processing. In particular, Skillman teaches [col. 4, line 37 - col. 9, line 29] data processing system (blackboard parallel processing system), comprising:

a plurality of event modules (knowledge source processors KSPs) each including code (knowledge source / event driven application) that generates an event data signal (trigger signal) representative of a particular event (upon occurrence of predefined condition / event);

a plurality of processing modules (knowledge source processors KSPs) distributed over the data processing system (fig. 4) each including code (knowledge source / event driven application) that provides processed data (output) [col. 6, lines 23-34];

a task module (blackboard control unit BCU 32, 106), selectively communicating (via communication module 38) with each of the plurality of event modules and the plurality of distributed processing modules (fig.s 3, 4), the task module including code for selecting and instantiating a plurality of instructions (initiate/sprawn a knowledge source process) that corresponds to the event data signal (trigger pattern is matched) and for executing the instructions (run processing) [col. 6, lines 23-34; col. 7, lines 5-30; col. 14, line 13 - col. 15, line 24].

Skillman further teaches dynamic information which includes statuses (execution is completed) of the distributed processing modules (KSPs), modification to the instructions (next instruction/task determined by event-pattern matching) and the processed data (output of KSP's processing); during execution of the instructions the task module provides and incorporates the dynamic information for real-time consideration thereof (processing output from a preceding KSP is transferred to database 34 of BCU which is then dispatched to the next KSP by scheduler 42); upon completion of the currently executing instruction (when a KSP completes its processing), the task module evaluates the incorporated dynamic information (trigger module of BCU determines if the output sent to database 34 matches a trigger definition) and selectively executes, based upon the incorporated dynamic information, the next instructions (if a match is found, scheduler sends a trigger message to start execution of the next KSP). See col. 5, lines 16-29; col. 7, lines 5-30; col. 15, lines 16-44; col. 18, lines 11-53.

In other words, Skillman controls the flow of an overall processing sequence by dispatching a trigger message and required data to the next KSP for processing. A Skillman's trigger message is the instruction(s) to the next KSP based on the processing output of the previous KSP in the sequence. Although Skillman does not call such instructions scripts, it would have been an obvious to do so. Skillman does not

explicitly teach that the instructions/script proceed from a first one to a second one of the distributed processing modules for processing a next instruction of the overall processing.

Fischer teaches distributed data processing, wherein instructions/script (traveling program) proceed from a first one to a second one of distributed processing modules (transmit itself to the next destination) for processing a next instruction in the sequence of the overall processing (to collect, edit and approve data). See col. 2, line 62 - col. 3, line 11. Execution of the next instruction is based on / incorporates the dynamic information (status 112, decision of next recipient) and processed data from previously processed instructions (data in PCB, VCB which are filled in by each processing stop and passed for subsequent processing). See col. 9, line 67 - col. 10, line 16; col. 10, line 40 - col. 12, line 36.

Given the teaching of Fischer, it would have been obvious for the instructions/script of Skillman to proceed from a first one to a second one of the distributed processing modules for processing a next instruction in the sequence of the overall processing.

The motivations to combine the teachings of Skillman and Fischer includes the following. Skillman discloses that each of the multiple KSP employs different architecture and needs to perform, during each communication, respective translation between a data format used by the KSP and a data format used by the blackboard control unit (col. 3, lines 59-64; col. 17, lines 3-62). This, to one of ordinary skill in the art, lacks efficiency and is error prone. Skillman desires efficiency (col. 2, line 65 – col. 3, line 3). On the other hand, Fischer provides a mechanism for data collection from multiple data producers, which reduces error of data format translation and improves efficiency (col. 2, line 62 – col. 3, line 2). Therefore, one of ordinary skill in the art would have been motivated to use the mechanism of Fischer to improve efficiency and reduce error associated with data format translation in Skillman.

As to claim 2, Skillman teaches the task module executes two or more of the plurality of scripts substantially simultaneously (parallel processing, col. 3, lines 11-32).

As to claim 3, Skillman teaches converter module (trigger module 46), in communication with the task module, including code that maps the event data signal to instructions/scripts upon reception of the event data signal by the task module [col. 6, lines 23-34].

As to claim 4, Skillman teaches the plurality of distributed processing modules (KSPs) provide event data signals (event), representative of particular events, to the task module. [col. 18, lines 42-53].

As to claims 5-7, Skillman teaches status monitoring module (trigger module, database module), in communication with the task module (BCU kernel), including code (global scheduler instructions in Table 1) that provides the status information to the task module including operating conditions of (start successful); in direct communication with the plurality of distributed processing modules (fig. 3); the status monitoring module stores data associated with the instance (output) of the selected instruction/script in an associated memory (database 36). [col. 13, line 50-54].

As to claims 8-9, Skillman teaches load balancing module (scheduler module), in communication with the task module (BCU kernel), including code that dynamically selects available ones of the plurality of distributed processing modules to perform processing (based on computational availability); in direct communication with the plurality of distributed processing modules (fig. 3). [col. 4, lines 3-8; col. 15, line 64 - col. 16, line 18].

As to claims 10-12, Skillman teaches bidirectionally and substantially simultaneously transmitting data between (parallel processing), resource management module (trigger module and scheduler module) for monitoring event data signal (trigger module) and for converting functionality (trigger message) in response to dynamic information regarding and available distributed processing modules to maximize (trigger least loaded KSP). [col. 6, lines 23-34; col. 7, lines 5-30; col. 14, line 13 - col. 15, line 24].

As to claims 13-19, Skillman teaches initiator modules / communication interfaces (communication module 38), regardless of native applications (KS robot, KS

vision), and protocols / communication interfaces (application interfaces, KSOSs) between various modules of the system [col. 5, line 56 - col. 6, line 34].

As to claim 20, Fischer teaches instructions/scripts (traveling program including its data structure) is preprogrammed to iteratively update its contents (loop to examine all FCB) [col. 9, line 67 - col. 10, line 16; col. 18, line 14 - col. 20, line 23]. Note discussion of claim 1 for motivation to combine.

As to claims 21-23, Skillman teaches storage module / persistent memory (database 34, 36).

As to claims 24-26, Skillman teaches script/instruction building module (trigger definitions defined in file TRIGGER.DEFS,), standard language interface ('C' programming language), GUI (display interface 148) [col. 12, lines 23-42].

As to claim 27, Skillman teaches dynamically updates and modifies (change KS in a dynamic fashion to meet requirements) scripts/instructions (KS) [it is noted that a general processing task, ie, scripts/instructions, of Skillman is implemented through knowledge sources and thus dynamic changes to the general processing task, ie, scripts/instructions, is represented by dynamic changes to the knowledge sources.]

As to claims 28-30, Skillman teaches protocols (inherent to communication interfaces 38), responder module (KSOS 112) to transmit response data from execution (pose output) [discussion of claim 1] and to convert format (translate between data formats) [col. 17, lines 53-62].

As to claims 31-32, Skillman teaches an administrative module (communication module 38, database module 34) that receives and presents data, and plurality of application peripherals (robot computer 134, vision computer 136) in communication with an associated one of the plurality of distributed processing modules / event modules (KSP 110b, KSP 110c) [col. 8, line 42 - col. 9, line 6].

As to claim 33, note discussion of claim 1, Skillman as modified further teaches resource management module (trigger module 46 and scheduler module 42) communicating with (fig.s 3, 4), for monitoring event data signals (output, event), for converting data processing functionality (dispatch/send another trigger signal) in response to dynamic information (trigger pattern match and completion status)

regarding the monitored event data signals (output, event) and the number of available distributed processing modules (based on computational capability and availability at the time of selection) to maximize number (run KSPs in parallel) of the distributed processing modules processing the event data signals [col. 4, lines 4-8; col. 6, lines 23-34; col. 7, lines 5-30; col. 14, line 13 - col. 15, line 24].

As to claim 34, note discussion of claim 1, Skillman teaches generating at least one event data signal (output, event) at one or more peripheral modules (robot computer 134, vision computer 136) and mapping the at least one event data signal to a selected script (portion of overall processing provided by KS) chosen from one or more scripts (KSs), each the one or more scripts having one or more instructions (processing capability) [col. 4, lines 4-8; col. 6, lines 23-34; col. 7, lines 5-30; col. 8, line 42 - col. 9, line 6; col. 14, line 13 - col. 15, line 24]; col. 19, lines 42-53.

As to claims 35-37, Skillman teaches communication interface (communication module 38, KSOS 110b, 110c), dynamically managing (scheduling, triggering) operating functions (KS) of the one or more peripheral modules (devices), response data signals as a result of the execution of selected instructions (previous processing output), transmitting the response data signals from the task module to (trigger message, local database/scheduler functions, table 2) selected ones of the one or more peripheral modules (KSPs) [col. 4, lines 4-8; col. 8, line 42 - col. 9, line 6; col. 13, lines 11-49; col. 14, line 13 - col. 15, line 24; col. 19, lines 42-53].

As to claims 38-40, refer to claims 32, 21 and 23, respectively, for discussions.

As to claims 41-43, Skillman teaches interface/communication between the task module and selected ones of the one or more peripheral modules (communication module 38, KSOS 110b, 110c), wherein a communication protocol would have been inherent for each KSP; interfacing with a plurality of the one or more peripheral modules substantially simultaneously (fig. 3 and 4); preceding instructions/processing to available processing modules (based on computational capability and availability at the time of selection). [col. 4, lines 4-8; col. 8, line 42 - col. 9, line 6; col. 13, lines 11-49; col. 14, line 13 - col. 15, line 24; col. 19, lines 42-53]. Note discussion of claim 1 with respect

to Fischer for preceding a script / instructions / portion of overall processing task to processing module(s).

As to claims 44 and 46, refer to claims 1 and 31, respectively, for discussions.

As to claim 45, Skillman as modified teaches the execution of the one or more instructions dynamically incorporates data gathered in previously executed instructions (Skillman, output from one KSP triggers is transferred to another KSP for processing, col. 7, lines 5-18) (Fischer, traveling program transmit itself and collected/attached data to next destination for processing, col. 9, line 67 - col. 10, line 17).

As to claim 60, note discussion of claim 1 and the equivalence of event data / event data signal. Skillman further teaches response profile (posted to global database) including results generated (processing output) and transmitting (send with trigger message) the response profile to the requesting event modules (next KSP in overall processing sequence) [col. 7, lines 5-30; col. 15, lines 16-44; col. 18, lines 11-53].

As to claim 61, Skillman teaches event data (predefined event, col. 18, lines 42-53).

As to claim 63, note discussion of claim 1. Skillman teaches first and second events because each of the KSPs is a source of event generation.

5. Claim 62 is rejected under 35 U.S.C. 103(a) as being unpatentable over Skillman et al and Fischer as applied to claim 60 and further in view of Waclawsky (U S Pat. 5,493,689).

As to claim 62, Waclawsky teaches tracing execution of instructions (trace, col. 1, lines 43-67). Continuing execution from a last traced instruction after failure modules from a last traced instruction is met by the well known roll-back protocol of transactional processing. Given the teaching of Waclawsky, it would have been obvious to include tracing and continuing steps into Skillman as modified. The motivation to combine the teachings includes the following. Skillman teaches select next processing module for processing based on load conditions [col. 15, lines 64-67], which, to one of ordinary skill in the art, would require a mechanism to collect and analyze load data from processing modules. Skillman does not provide such a mechanism. Waclawsky on the other hand,

provides a mechanism for collecting and analyzing load data from processing modules (col. 1, line 43 - col. 2, line 62). Therefore, one of ordinary skill in the art would have been motivated to use the mechanism of Waclawsky to collect and analyze load data so that selection of next KSP can be made based on load conditions.

6. Claims 1-46 and 60, 61, 63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Skillman et al (US Pat. 5,506,999) in view of Walsh (US Pat. 6,233,601)

As to claim 1, Skillman teaches a plurality of processing units each executing a portion of an overall application and a centralized event-driven control to integrate the distributed processing. In particular, Skillman teaches [col. 4, line 37 - col. 9, line 29] data processing system (blackboard parallel processing system), comprising:

a plurality of event modules (knowledge source processors KSPs) each including code (knowledge source / event driven application) that generates an event data signal (trigger signal) representative of a particular event (upon occurrence of predefined condition / event);

a plurality of processing modules (knowledge source processors KSPs) distributed over the data processing system (fig. 4) each including code (knowledge source / event driven application) that provides processed data (output) [col. 6, lines 23-34];

a task module (blackboard control unit BCU 32, 106), selectively communicating (via communication module 38) with each of the plurality of event modules and the plurality of distributed processing modules (fig.s 3, 4), the task module including code for selecting and instantiating a plurality of instructions (initiate/sprawn a knowledge source process) that corresponds to the event data signal (trigger pattern is matched) and for executing the instructions (run processing) [col. 6, lines 23-34; col. 7, lines 5-30; col. 14, line 13 - col. 15, line 24].

Skillman further teaches dynamic information which includes statuses (execution is completed) of the distributed processing modules (KSPs), modification to the instructions (next instruction/task determined by event-pattern matching) and the

processed data (output of KSP's processing); during execution of the instructions the task module provides and incorporates the dynamic information for real-time consideration thereof (processing output from a preceding KSP is transferred to database 34 of BCU which is then dispatched to the next KSP by scheduler 42); upon completion of the currently executing instruction (when a KSP completes its processing), the task module evaluates the incorporated dynamic information (trigger module of BCU determines if the output sent to database 34 matches a trigger definition) and selectively executes, based upon the incorporated dynamic information, the next instructions (if a match is found, scheduler sends a trigger message to start execution of the next KSP). See col. 5, lines 16-29; col. 7, lines 5-30; col. 15, lines 16-44; col. 18, lines 11-53.

In other words, Skillman controls the flow of an overall processing sequence by dispatching a trigger message and required data to the next KSP for processing. A Skillman's trigger message is the instruction(s) to the next KSP based on the processing output of the previous KSP in the sequence. Although Skillman does not call such instructions scripts, it would have been an obvious to do so. Skillman does not explicitly teach that the instructions/script proceed from a first one to a second one of the distributed processing modules for processing a next instruction of the overall processing.

Walsh teaches distributed data processing, wherein instructions/script (mobile agent object 10) proceed from a first one to a second one of distributed processing modules (migrate to another computer) for processing a next instruction in the sequence of the overall processing (method 2 of server 2, method 3 of server 3, ..., as defined by the itinerary data structure). See col. 2, lines 41-59; col. 5, lines 27-36. Given the teaching of Walsh, it would have been obvious for the instructions/script of Skillman to proceed from a first one to a second one of the distributed processing modules for processing a next instruction in the sequence of the overall processing. One of ordinary skill in the art would have been motivated to combine the teachings of Skillman with Walsh because this would have reduced network traffic incurred by transmitting unneeded instructions/code around the network (Walsh, col. 2, lines 16-40),

thus speeding up useful communications, as desired by Skillman (col. 2, line 62 – col. 3, line 2).

As to claim 2, Skillman teaches the task module executes two or more of the plurality of scripts substantially simultaneously (parallel processing, col. 3, lines 11-32).

As to claim 3, Skillman teaches converter module (trigger module 46), in communication with the task module, including code that maps the event data signal to instructions/scripts upon reception of the event data signal by the task module [col. 6, lines 23-34].

As to claim 4, Skillman teaches the plurality of distributed processing modules (KSPs) provide event data signals (event), representative of particular events, to the task module. [col. 18, lines 42-53].

As to claims 5-7, Skillman teaches status monitoring module (trigger module, database module), in communication with the task module (BCU kernel), including code (global scheduler instructions in Table 1) that provides the status information to the task module including operating conditions of (start successful); in direct communication with the plurality of distributed processing modules (fig. 3); the status monitoring module stores data associated with the instance (output) of the selected instruction/script in an associated memory (database 36). [col. 13, line 50-54].

As to claims 8-9, Skillman teaches load balancing module (scheduler module), in communication with the task module (BCU kernel), including code that dynamically selects available ones of the plurality of distributed processing modules to perform processing (based on computational availability); in direct communication with the plurality of distributed processing modules (fig. 3). [col. 4, lines 3-8; col. 15, line 64 - col. 16, line 18].

As to claims 10-12, Skillman teaches bidirectionally and substantially simultaneously transmitting data between (parallel processing), resource management module (trigger module and scheduler module) for monitoring event data signal (trigger module) and for converting functionality (trigger message) in response to dynamic information regarding and available distributed processing modules to maximize (trigger

least loaded KSP). [col. 6, lines 23-34; col. 7, lines 5-30; col. 14, line 13 - col. 15, line 24].

As to claims 13-19, Skillman teaches initiator modules / communication interfaces (communication module 38), regardless of native applications (KS robot, KS vision), and protocols / communication interfaces (application interfaces, KSOSs) between various modules of the system [col. 5, line 56 - col. 6, line 34].

As to claim 20, Walsh teaches instructions/scripts is preprogrammed to iteratively update its contents (supplemented with code from home codebase, col. 6, lines 31-65). Note discussion of claim 1 for a motivation to combine.

As to claims 21-23, Skillman teaches storage module / persistent memory (database 34, 36).

As to claims 24-26, Skillman teaches script/instruction building module (trigger definitions defined in file TRIGGER.DEFS,), standard language interface ('C' programming language), GUI (display interface 148) [col. 12, lines 23-42].

As to claim 27, Skillman teaches dynamically updates and modifies (change KS in a dynamic fashion to meet requirements) scripts/instructions (KS) [it is noted that a general processing task, ie, scripts/instructions, of Skillman is implemented through knowledge sources and thus dynamic changes to the general processing task, ie, scripts/instructions, is represented by dynamic changes to the knowledge sources.]

As to claims 28-30, Skillman teaches protocols (inherent to communication interfaces 38), responder module (KSOS 112) to transmit response data from execution (pose output) [discussion of claim 1] and to convert format (translate between data formats) [col. 17, lines 53-62].

As to claims 31-32, Skillman teaches an administrative module (communication module 38, database module 34) that receives and presents data, and plurality of application peripherals (robot computer 134, vision computer 136) in communication with an associated one of the plurality of distributed processing modules / event modules (KSP 110b, KSP 110c) [col. 8, line 42 - col. 9, line 6].

As to claim 33, note discussion of claim 1, Skillman as modified further teaches resource management module (trigger module 46 and scheduler module 42)

communicating with (fig.s 3, 4), for monitoring event data signals (output, event), for converting data processing functionality (dispatch/send another trigger signal) in response to dynamic information (trigger pattern match and completion status) regarding the monitored event data signals (output, event) and the number of available distributed processing modules (based on computational capability and availability at the time of selection) to maximize number (run KSPs in parallel) of the distributed processing modules processing the event data signals [col. 4, lines 4-8; col. 6, lines 23-34; col. 7, lines 5-30; col. 14, line 13 - col. 15, line 24].

As to claim 34, note discussion of claim 1, Skillman teaches generating at least one event data signal (output, event) at one or more peripheral modules (robot computer 134, vision computer 136) and mapping the at least one event data signal to a selected script (portion of overall processing provided by KS) chosen from one or more scripts (KSs), each the one or more scripts having one or more instructions (processing capability) [col. 4, lines 4-8; col. 6, lines 23-34; col. 7, lines 5-30; col. 8, line 42 - col. 9, line 6; col. 14, line 13 - col. 15, line 24]; col. 19, lines 42-53.

As to claims 35-37, Skillman teaches communication interface (communication module 38, KSOS 110b, 110c), dynamically managing (scheduling, triggering) operating functions (KS) of the one or more peripheral modules (devices), response data signals as a result of the execution of selected instructions (previous processing output), transmitting the response data signals from the task module to (trigger message, local database/scheduler functions, table 2) selected ones of the one or more peripheral modules (KSPs) [col. 4, lines 4-8; col. 8, line 42 - col. 9, line 6; col. 13, lines 11-49; col. 14, line 13 - col. 15, line 24; col. 19, lines 42-53].

As to claims 38-40, refer to claims 32, 21 and 23, respectively, for discussions.

As to claims 41-43, Skillman teaches interface/communication between the task module and selected ones of the one or more peripheral modules (communication module 38, KSOS 110b, 110c), wherein a communication protocol would have been inherent for each KSP; interfacing with a plurality of the one or more peripheral modules substantially simultaneously (fig. 3 and 4); preceding instructions/processing to available processing modules (based on computational capability and availability at the

time of selection). [col. 4, lines 4-8; col. 8, line 42 - col. 9, line 6; col. 13, lines 11-49; col. 14, line 13 - col. 15, line 24; col. 19, lines 42-53]. Note discussion of claim 1 with respect to Walsh for proceeding a script / instructions / portion of overall processing task to processing module(s).

As to claims 44 and 46, refer to claims 1 and 31, respectively, for discussions.

As to claim 45, Skillman as modified teaches the execution of the one or more instructions dynamically incorporates data gathered in previously executed instructions (Skillman, output from one KSP triggers is transferred to another KSP for processing, col. 7, lines 5-18) (Walsh, migrate to another computer/next agent manager, col. 2, lines 41-59; col. 5, lines 27-36.).

As to claim 60, note discussion of claim 1 and the equivalence of event data / event data signal. Skillman further teaches response profile (posted to global database) including results generated (processing output) and transmitting (send with trigger message) the response profile to the requesting event modules (next KSP in overall processing sequence) [col. 7, lines 5-30; col. 15, lines 16-44; col. 18, lines 11-53].

As to claim 61, Skillman teaches event data (predefined event, col. 18, lines 42-53).

As to claim 63, note discussion of claim 1. Skillman teaches first and second events because each of the KSPs is a source of event generation.

7. Claim 62 is rejected under 35 U.S.C. 103(a) as being unpatentable over Skillman et al and Walsh as applied to claim 60 and further in view of Waclawsky (U S Pat. 5,493,689).

As to claim 62, Waclawsky teaches tracing execution of instructions (trace, col. 1, lines 43-67). Continuing execution from a last traced instruction after failure modules from a last traced instruction is met by the well known roll-back protocol of transactional processing. Given the teaching of Waclawsky, it would have been obvious to include tracing and continuing steps into Skillman as modified. The motivation to combine the teachings includes the following. Skillman teaches select next processing module for processing based on load conditions [col. 15, lines 64-67], which, to one of ordinary skill

in the art, would require a mechanism to collect and analyze load data from processing modules. Skillman does not provide such a mechanism. Waclawsky on the other hand, provides a mechanism for collecting and analyzing load data from processing modules (col. 1, line 43 - col. 2, line 62). Therefore, one of ordinary skill in the art would have been motivated to use the mechanism of Waclawsky to collect and analyze load data so that selection of next KSP can be made based on load conditions.

8. Applicant's arguments filed 11/26/2003 have been considered but are moot in view of the new ground(s) of rejection.

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (703) 305-9657. A voice mail service is also available at this number. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7238 for After Final communications, (703) 746-7239 for Official communications and (703) 746-7240 for Non-Official/Draft communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Sue Lao Sue Lao

February 2, 2004